Lecture 3:
BERT: Bidirectional Encoder Representations from Transformers

Dr. Mohamed Reda Bouadjenek

School of Information Technology,
Faculty of Sci Eng & Built Env

reda.bouadjenek@deakin.edu.au

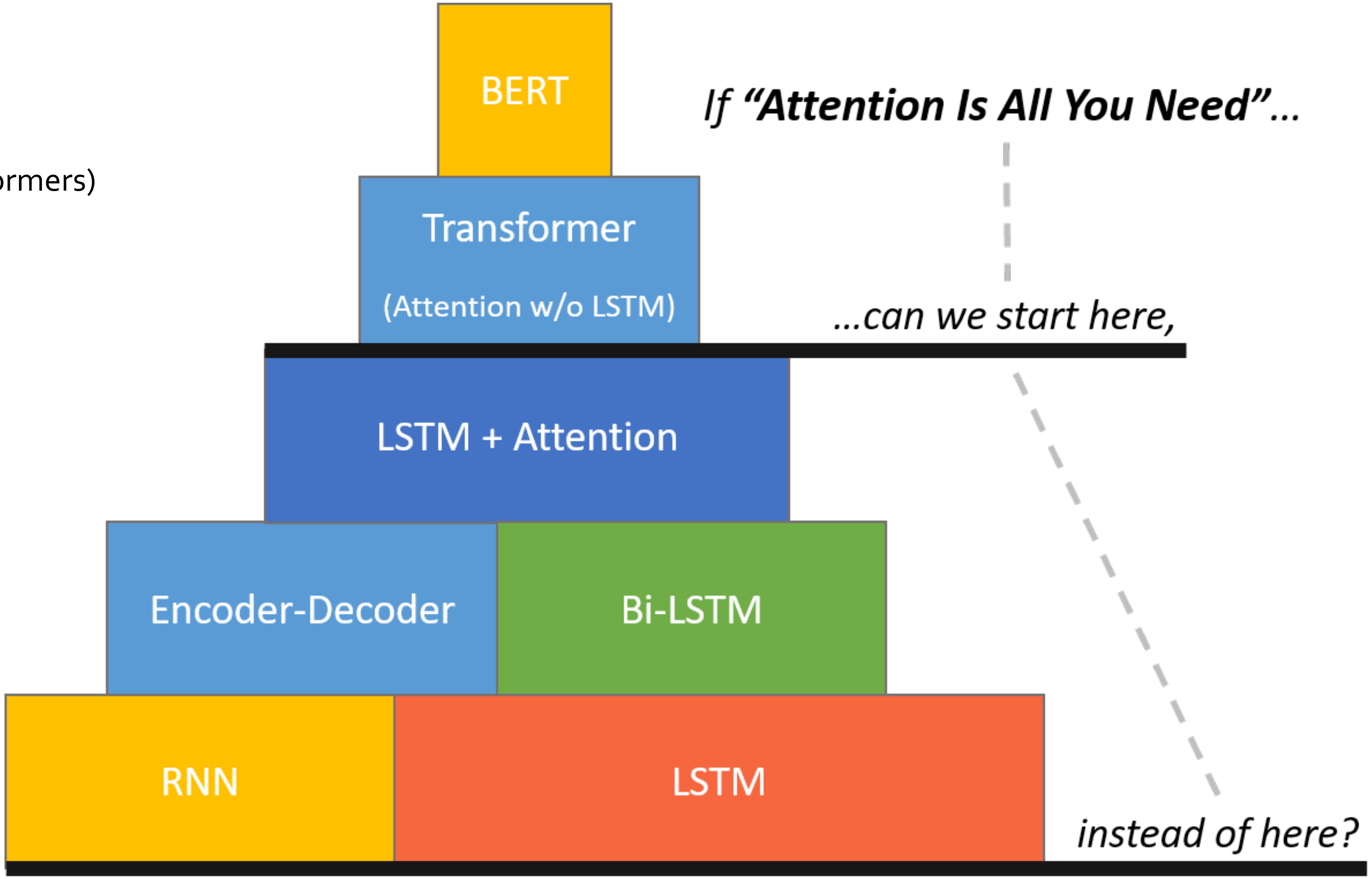
June 4, 2020

DEAKIN UNIVERSITY

The BERT Mountain!

By Chris McCormick

(Bidirectional Encoder Representations from Transformers)



Outline

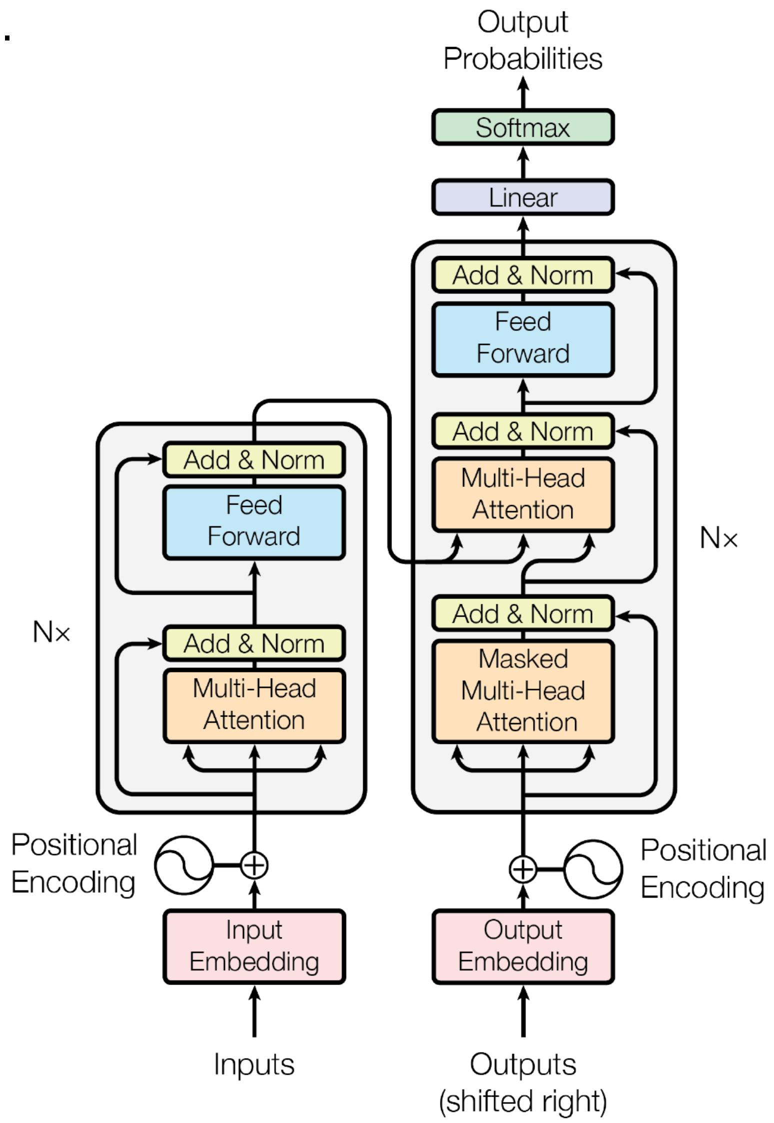
- **Quick Recap:** Transformers (previous talk!)
- BERT
    - Architecture
    - Input Representation
    - Pre-training procedure: Masked LM and Next Sentence Prediction
    - Fine-tuning procedure
- Experiments
- Conclusions

Quick Recap: Transformers

Ashish Vaswani et. Attention Is All You Need. NIPS 2017.

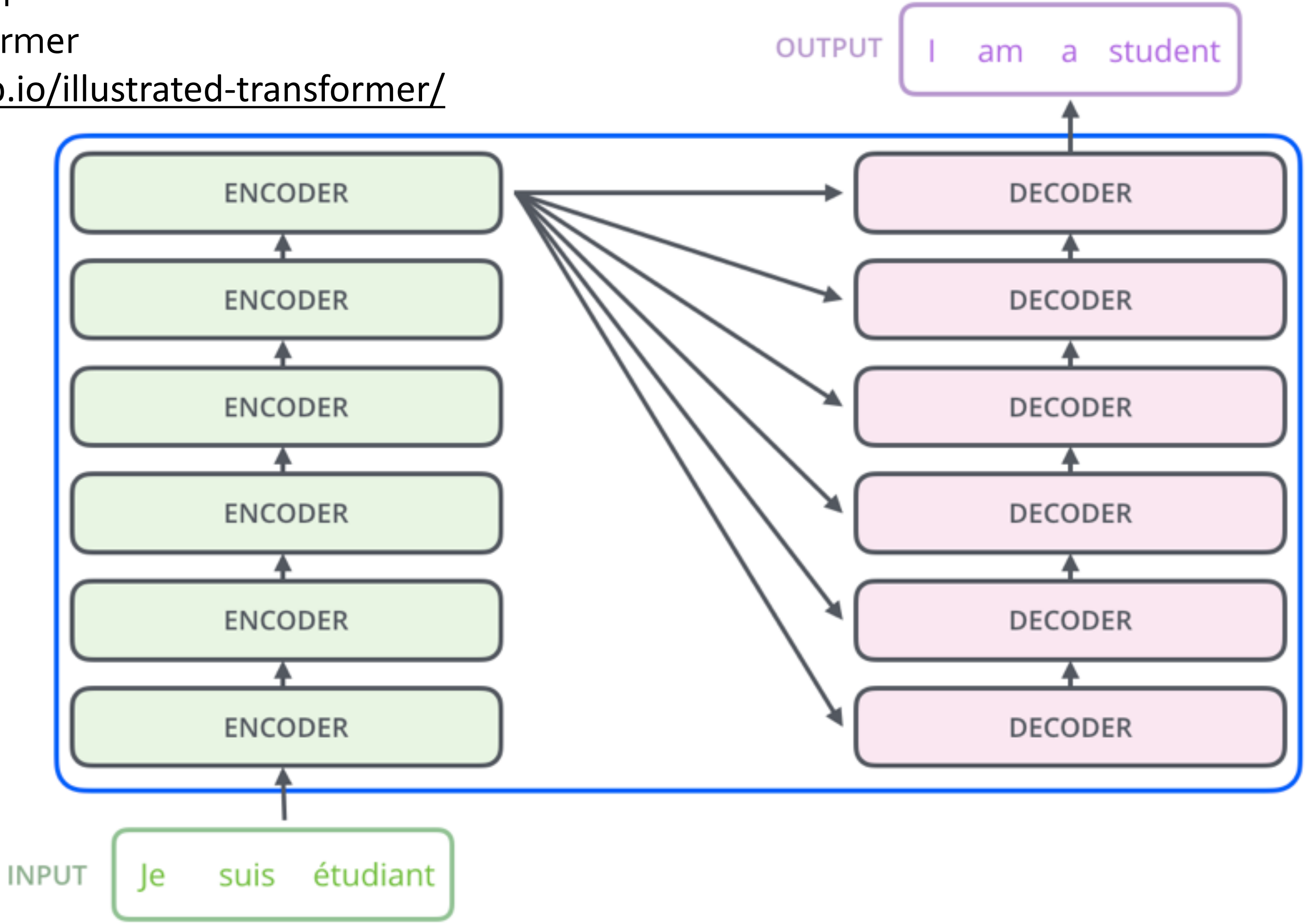
Encoder-Decoder

Ashish Vaswani et. Attention Is All You Need.
NIPS 2017.



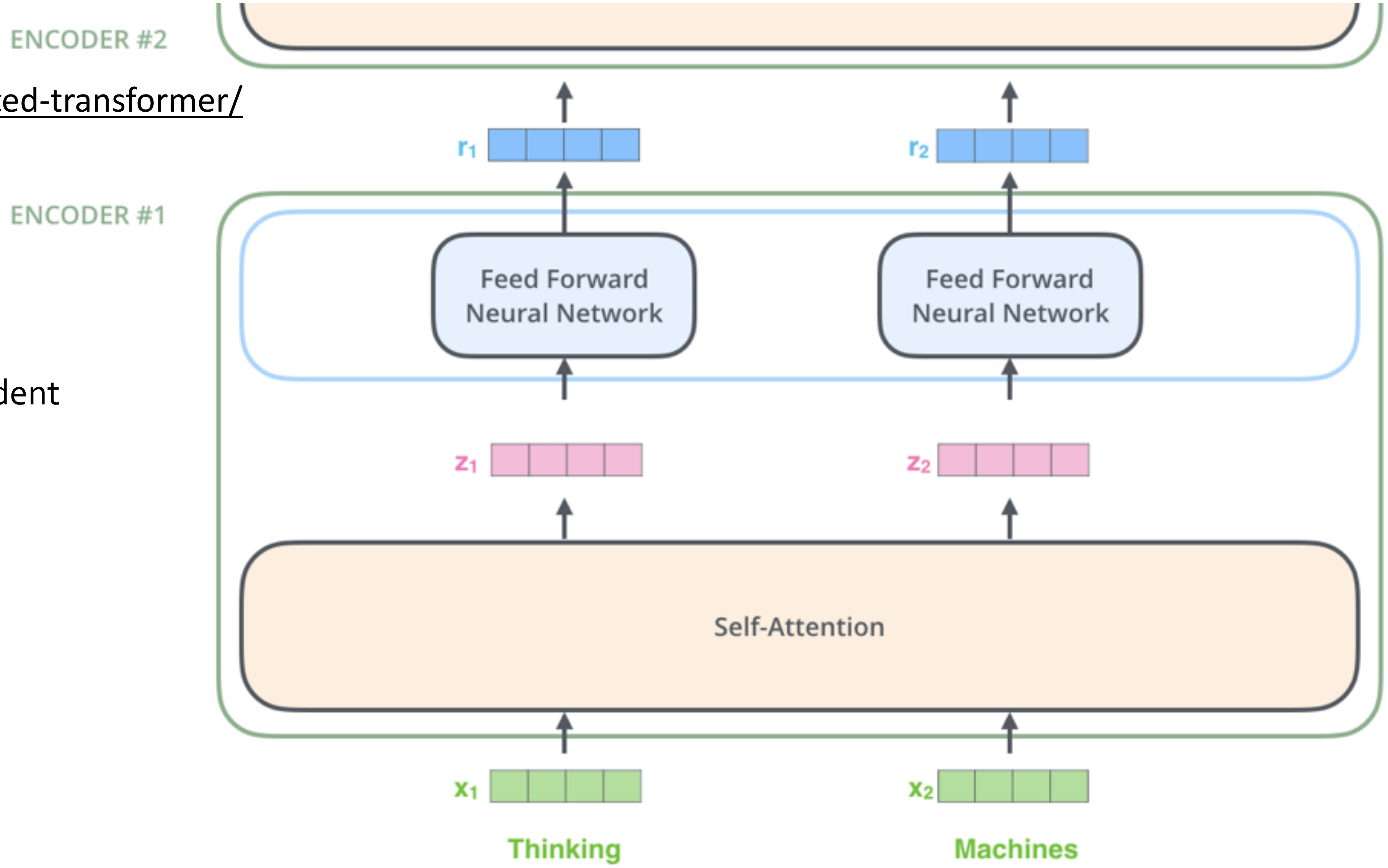
Encoder-Decoder

Figure by: Jay Alammar
The Illustrated Transformer
http://jalammar.github.io/illustrated-transformer/



Encoder

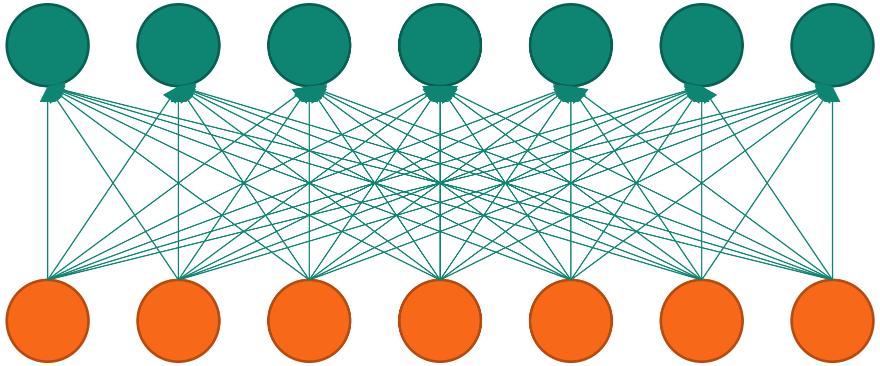
Figure by: Jay Alammar
The Illustrated Transformer
http://jalammar.github.io/illustrated-transformer/



Note: The FFNN is independent for each word.
Hence can be parallelized.

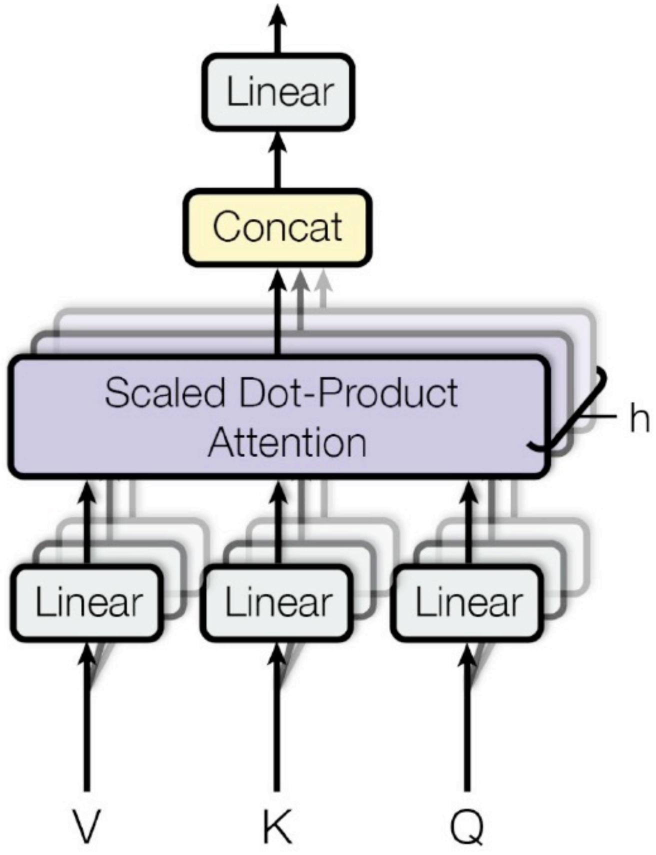
Self-Attention Mechanism

- All-to-all comparison.
  - Each layer is $O(N^2)$ for sequence of length N - **self attention**.
- Every output is a weighted sum of every input.
  - The weighting is a function to learn.



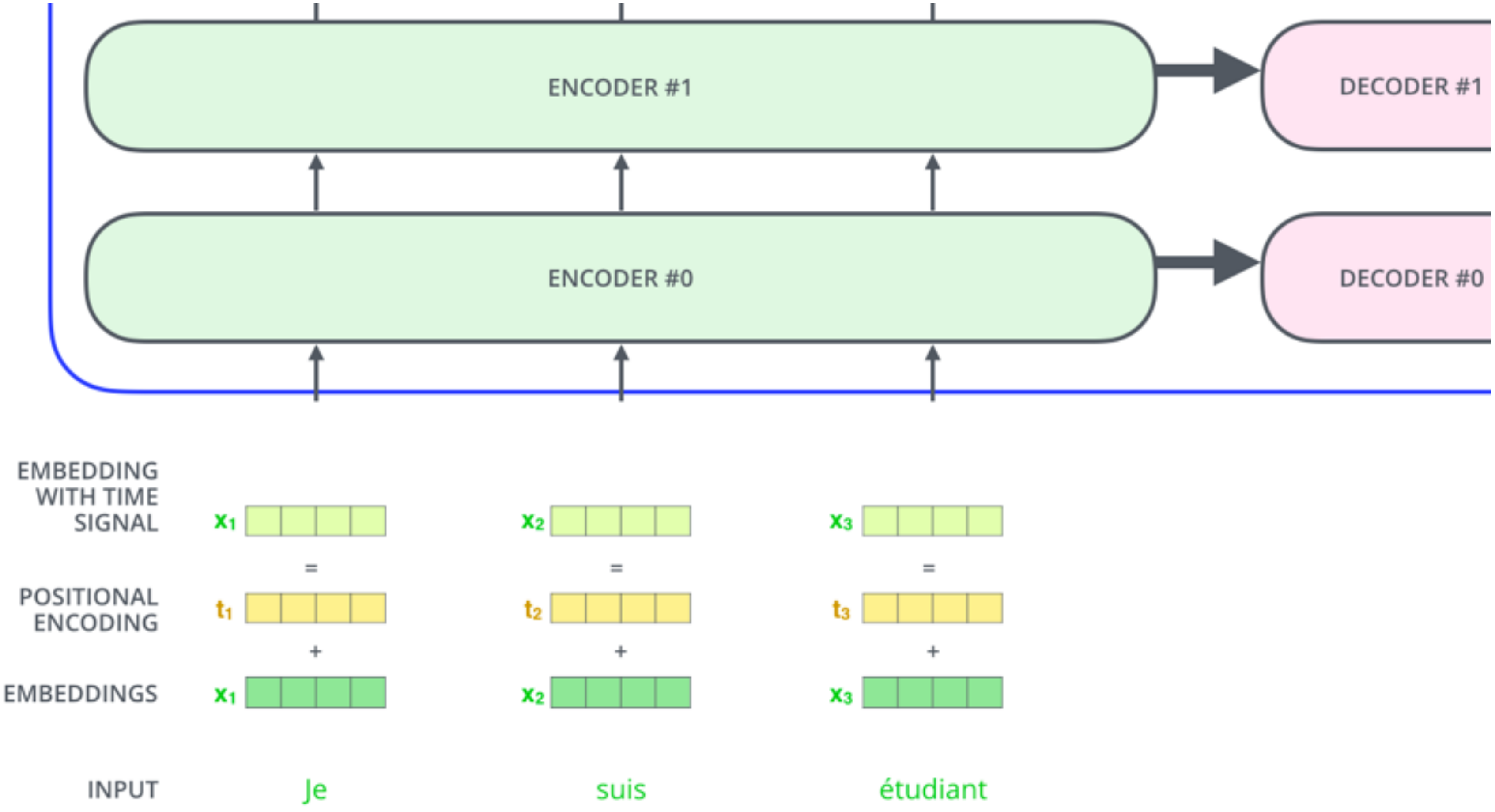
Multi-Headed Attention

- Clever, important innovation.
  - Not that hard.
- Just do that same thing 8 times with different Q,K,V matrices.
- Let the network learn 8 different semantic meanings of attention.
  - E.g., One grammar, one for vocabulary, one for conjugation, etc.
  - Very flexible mechanism for sequence processing.



Position encoding

Figure by: Jay Alammar
The Illustrated Transformer
http://jalammar.github.io/illustrated-transformer/

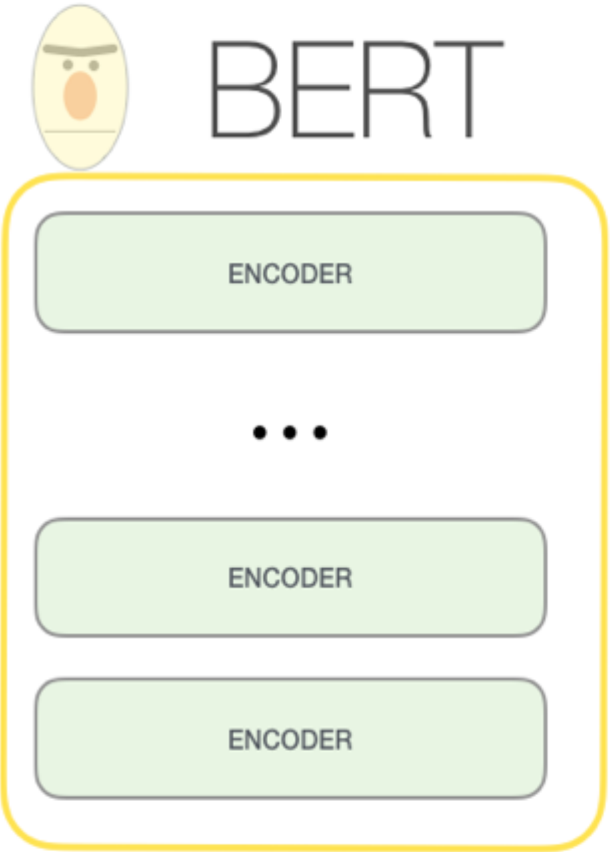


BERT

Bidirectional Encoder Representations from **Transformers**

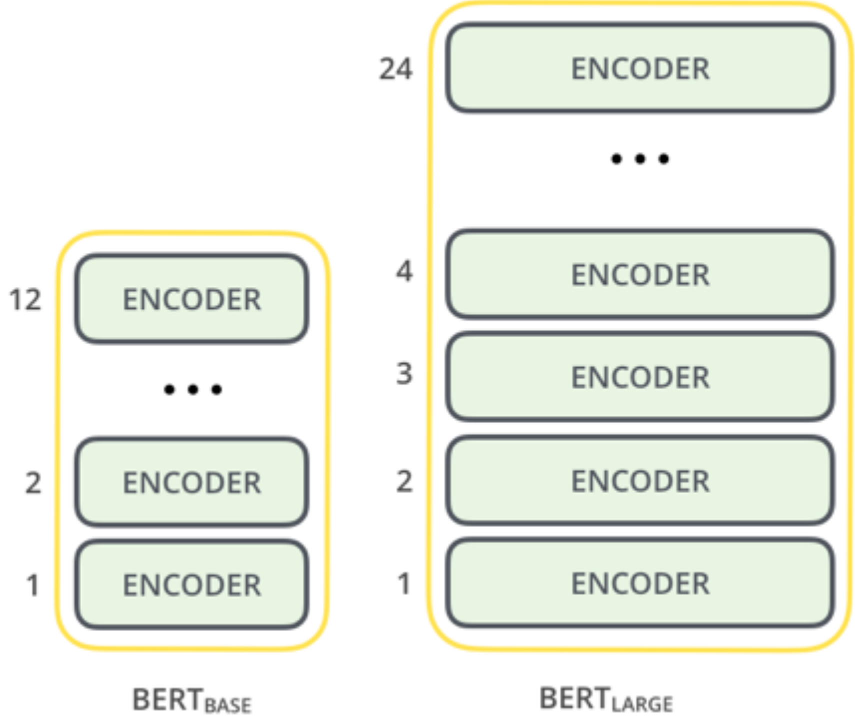
Model architecture

- Transformers is an **Encoder-Decoder** architecture
    - A transformer uses Encoder stack to model input, and uses Decoder stack to model output (using input information from encoder side).
- BERT
    - If we are only interested in training a language model for the input for some other tasks, then we do not need the Decoder of the transformer, that gives us BERT.

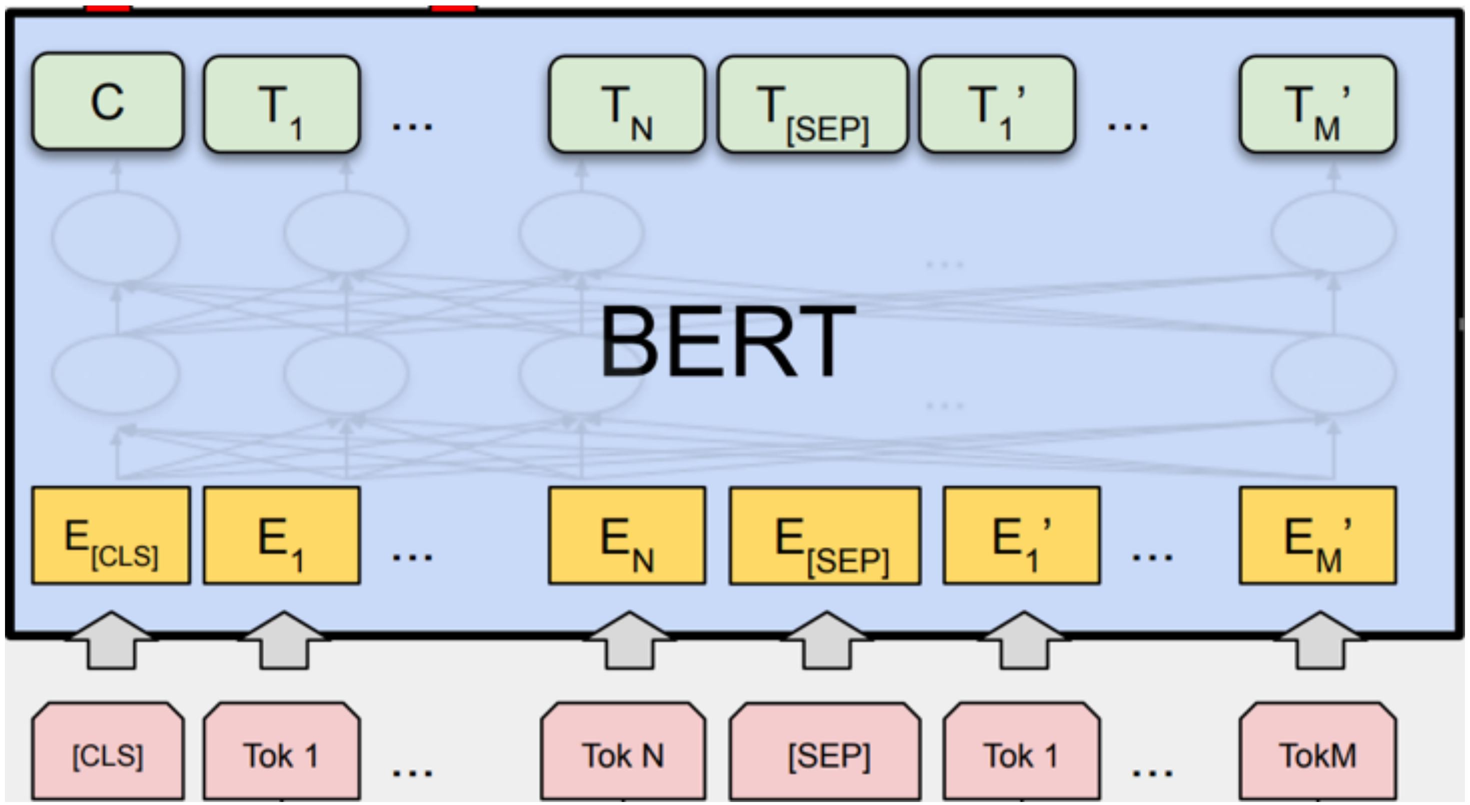


Model architecture

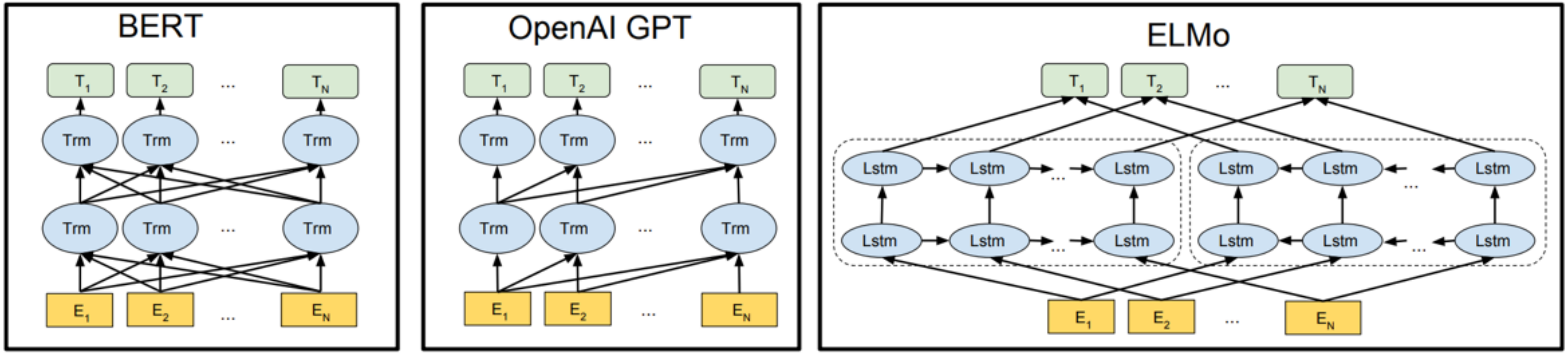
- Two models with different sizes were investigated
  - $\text{BERT}_{\text{BASE}}$ (cased and uncased) : L=12, H=768, A=12, Total Parameters=110M
    - (L: number of layers (Transformer blocks), H is the hidden size, A: the number of self-attention heads)
  - $\text{BERT}_{\text{LARGE}}$: L=24, H=1024, A=16, Total Parameters=340M



BERT in Action

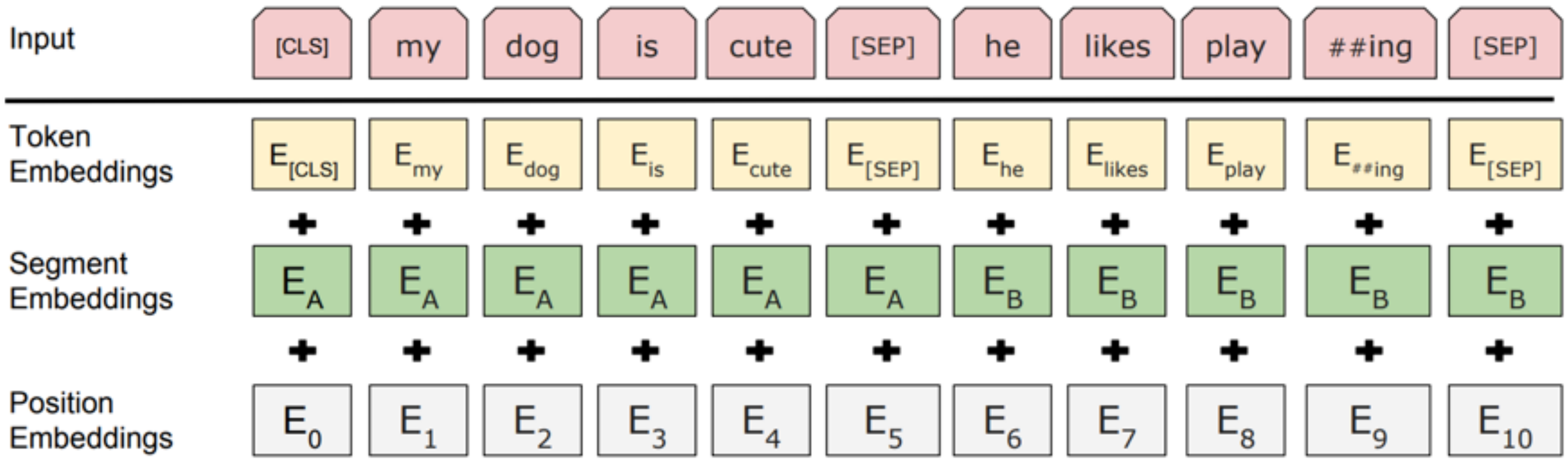


Differences in pre-training model architectures: BERT, OpenAI GPT, and ELMo



- GPT is built using transformer decoder blocks. BERT, on the other hand, uses transformer encoder blocks.
- GPT is auto-regressive in nature. BERT is not.
  - In losing auto-regression, BERT gained the ability to incorporate the context on both sides of a word to gain better results.

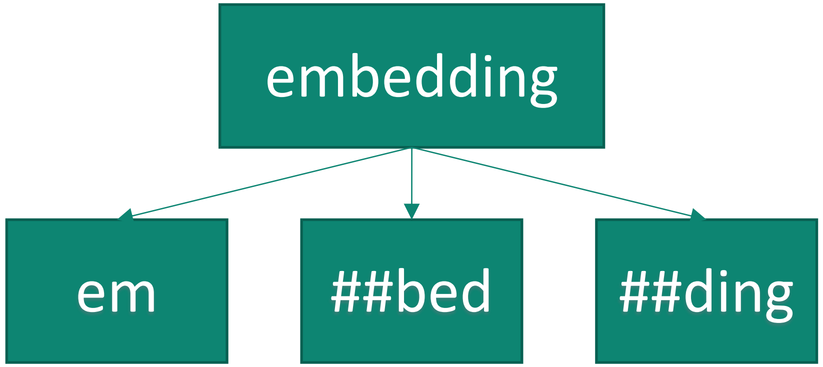
Input Representation

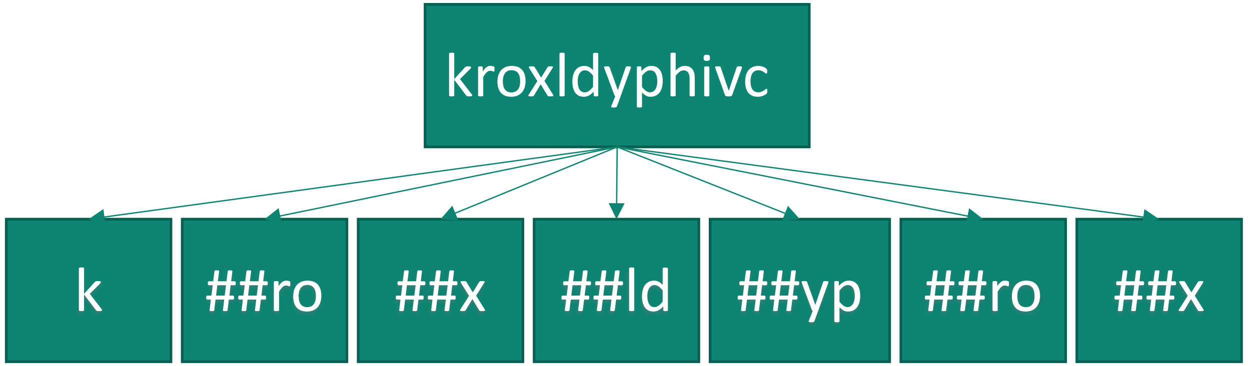


- **Token Embeddings:** Use pretrained WordPiece embeddings.
- **Segment Embeddings (Optional):** Added sentence embedding to every tokens of each sentence.
- **Position Embeddings:** Use learned Position Embeddings.
- Use **[CLS]** for the classification tasks.
- Separate sentences by using a special token **[SEP]**.

BERT’s Vocabulary

- BERT is pre-trained ➔ Vocabulary is fixed
  - The vocabulary contains **30,522** tokens.
  - How to deal with unknown words?
- Break down **unknown words** into **subwords**:
  - Using the **wordpiece** model.
- A subword exists for every character.
  - 2 types of subwords
    - All subwords start with “##”...
    - Except for the first subword in a word





Pre-training
Task#1: Masked LM

- 15% of the words are masked at random
  - and the task is to predict the masked words based on its left and right context
- Not all tokens were masked in the same way (example sentence “My dog is hairy”)
  - 80% were replaced by the <MASK> token: “My dog is <MASK>”
  - 10% were replaced by a random token: “My dog is apple”
  - 10% were left intact: “My dog is hairy”

Pre-training
Task#2: Next Sentence Prediction

- Motivation
  - Many downstream tasks are based on understanding the relationship between two text sentences
    - Question Answering (QA) and Natural Language Inference (NLI)
  - Language modeling does not directly capture that relationship
- The task is pre-training binarized next sentence prediction task

Input = **[CLS]** the man went to **[MASK]** store **[SEP]** he bought a gallon **[MASK]** milk **[SEP]**

Label = isNext

Input = **[CLS]** the man **[MASK]** to the store **[SEP]** penguin **[MASK]** are flight ##less birds **[SEP]**

Label = NotNext

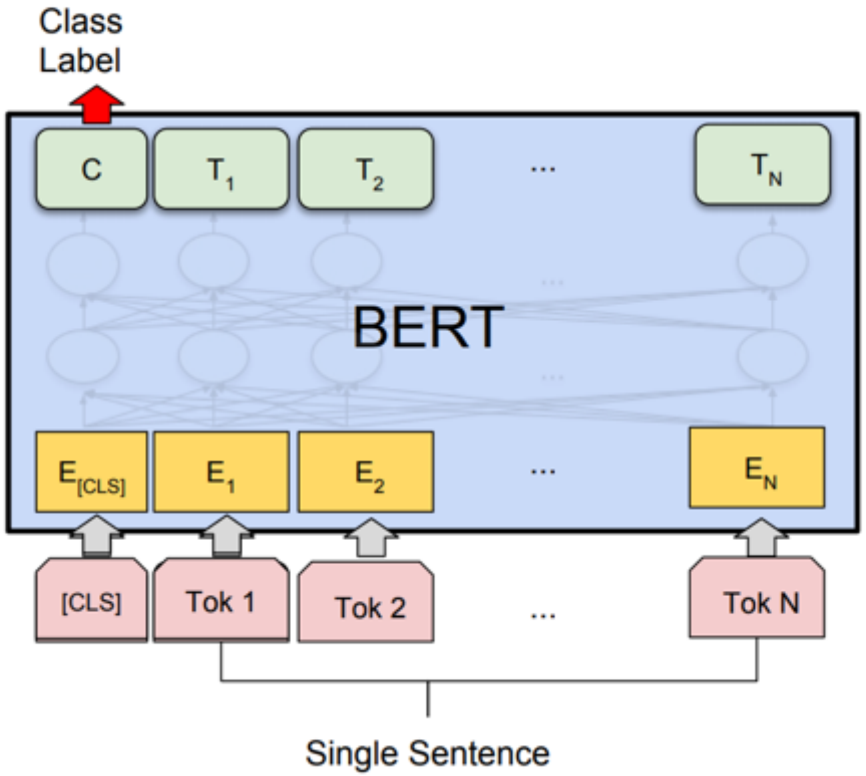
Pre-training procedure

- Training data: BooksCorpus (800M words) + English Wikipedia (2.5B words).
- To generate each training input sequences: sample two spans of text (A and B) from the corpus.
  - The combined length is ≤ 500 tokens.
  - 50% B is the actual next sentence that follows A and 50% of the time it is a random sentence from the corpus.
- The training loss is the sum of the mean masked LM likelihood and the mean next sentence prediction likelihood.

Fine-tuning procedure 1: Classification

- For sequence-level classification task
  - Obtain the representation of the input sequence by using the final hidden state (hidden state at the position of the special token [CLS]) $C \in R^H$
  - Just add a classification layer and use **softmax** to calculate label probabilities. Parameters $W \in R^{K \times H}$

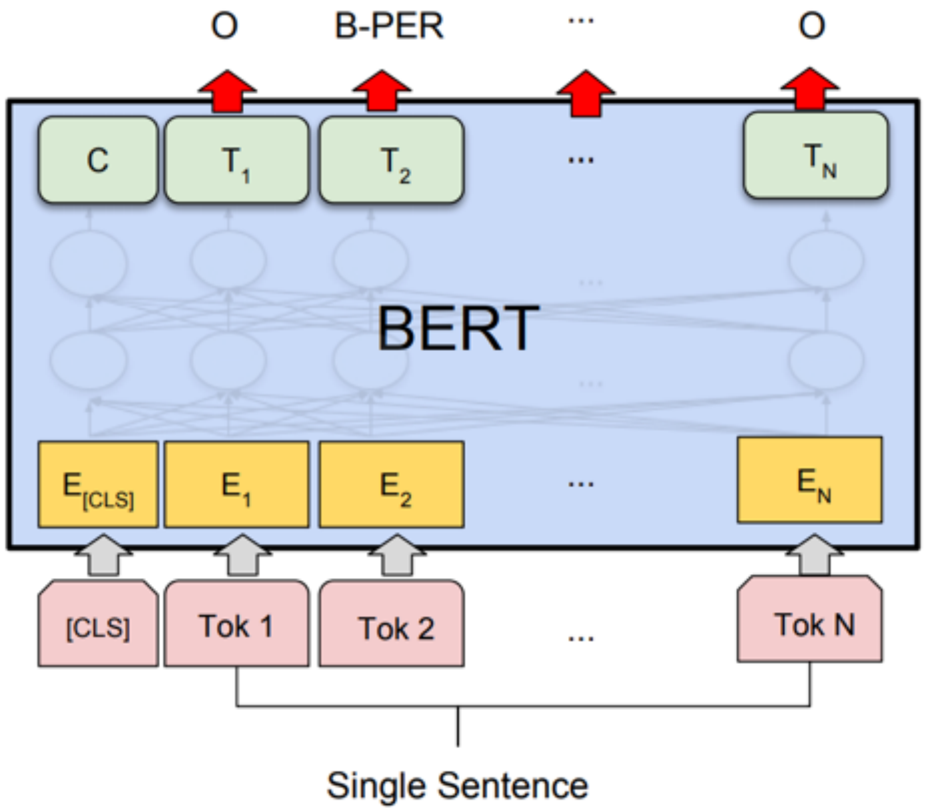
$$P = \textbf{softmax}(CW^T)$$



Fine-tuning procedure 2: Named Entity Recognition

- Feed the final hidden representation $\boldsymbol{T_i} \in \boldsymbol{R^H}$ for each token $\boldsymbol{i}$ into a classification layer for the tagset.
- To make the task compatible with WordPiece tokenization
  - Predict the tag for the first sub-token of a word
  - No prediction is made for X

Jim	**Hen**	**##son**	**was**	**a**	**puppet**	**#er**
I-PER	**I-PER**	**X**	**0**	**0**	**0**	**X**



Fine-tuning procedure 3: Query Answering 1/2

- **Input Question:**

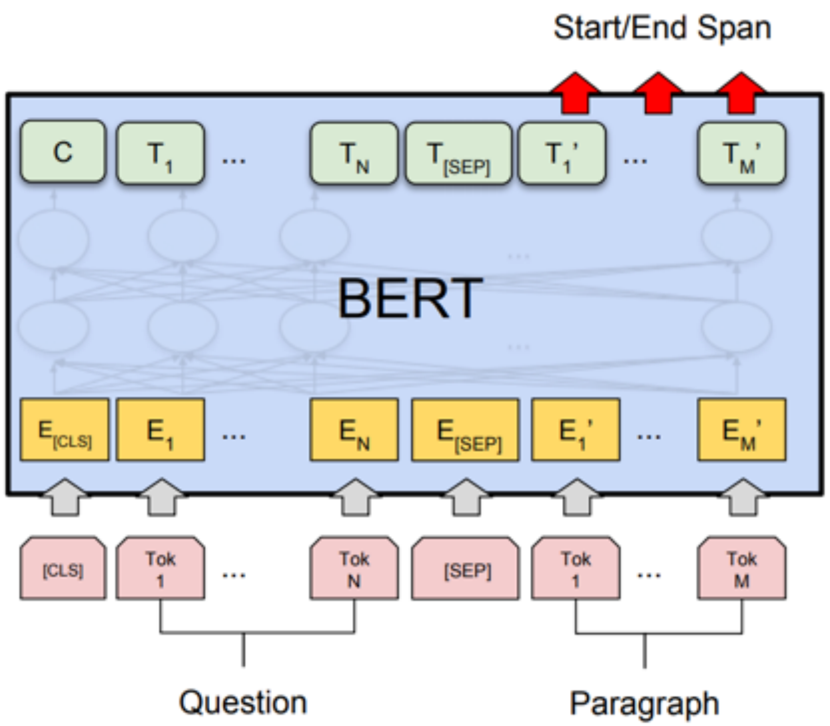
Where do water droplets collide with ice crystals to form precipitation?

- **Input Paragraph:**

…. Precipitation forms as smaller dropletscoalesce via collision with other rain dropsor ice crystals within a cloud. …

- **Output Answer:**

within a cloud

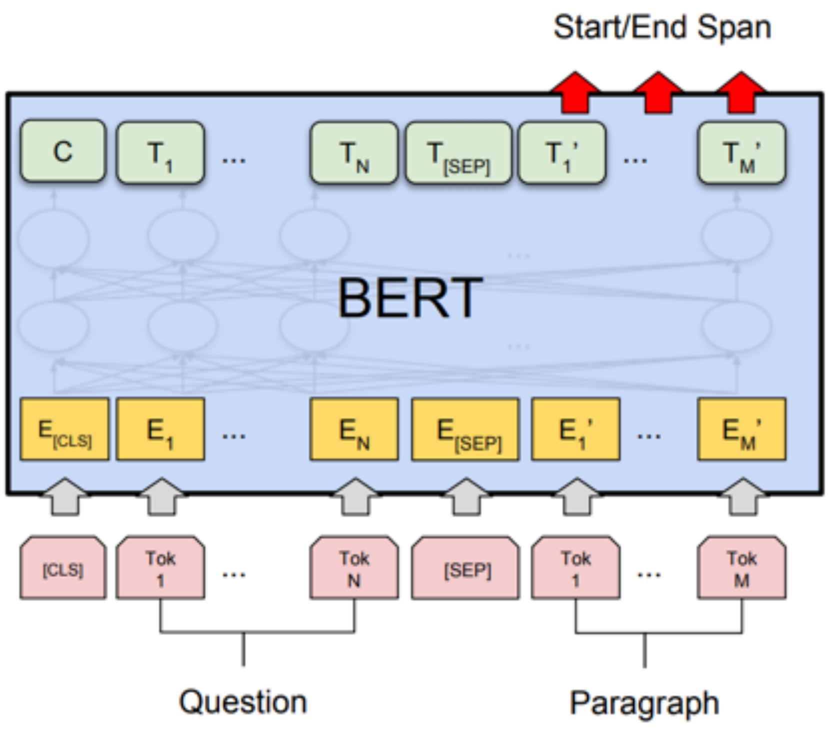


Fine-tuning procedure 3: Query Answering 2/2

- Represent the input question and paragraph as a single packed sequence.
  - The question uses the **A** embedding and the paragraph uses the **B** embedding.
- New parameters to be learned in fine-tuning are start vector $S \in \boldsymbol{R}^{\boldsymbol{H}}$ and end vector $\boldsymbol{E} \in \boldsymbol{R}^{\boldsymbol{H}}$.
- Calculate the probability of word & being the start of the answer span:

$$\boldsymbol{P_{Start}} = \boldsymbol{Softmax(ST^T)} \text{ and } \boldsymbol{P_{end}} = \boldsymbol{Softmax(ET^T)}$$

- The training objective is the log-likelihood the correct and end positions.



Experiments

Experiments

- GLUE (General Language Understanding Evaluation) benchmark
  - Distribute canonical Train, Dev and Test splits
  - Labels for Test set are not provided
- Datasets in GLUE:
  - MNLI: Multi-Genre Natural Language Inference
  - QQP: Quora Question Pairs
  - QNLI: Question Natural Language Inference
  - SST-2: Stanford Sentiment Treebank
  - CoLA: The corpus of Linguistic Acceptability
  - STS-B: The Semantic Textual Similarity Benchmark
  - MRPC: Microsoft Research Paraphrase Corpus
  - RTE: Recognizing Textual Entailment
  - WNLI: Winograd NLI

GLUE Results

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (https://gluebenchmark.com/leaderboard). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

SQuAD: The Stanford Question Answering Dataset

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	-	-	82.3	91.2
#1 Ensemble - nlnet	-	-	86.0	91.7
#2 Ensemble - QANet	-	-	84.5	90.5
Published				
BiDAF+ELMo (Single)	-	85.6	-	85.8
R.M. Reader (Ensemble)	81.2	87.9	82.3	88.5
Ours				
$\text{BERT}_{\text{BASE}}$ (Single)	80.8	88.5	-	-
$\text{BERT}_{\text{LARGE}}$ (Single)	84.1	90.9	-	-
$\text{BERT}_{\text{LARGE}}$ (Ensemble)	85.8	91.8	-	-
$\text{BERT}_{\text{LARGE}}$ (Sgl.+TriviaQA)	**84.2**	**91.1**	**85.1**	**91.8**
$\text{BERT}_{\text{LARGE}}$ (Ens.+TriviaQA)	**86.2**	**92.2**	**87.4**	**93.2**

Table 2: SQuAD 1.1 results. The BERT ensemble is 7x systems which use different pre-training checkpoints and fine-tuning seeds.

System	Dev EM	Dev F1	Test EM	Test F1
Top Leaderboard Systems (Dec 10th, 2018)				
Human	86.3	89.0	86.9	89.5
#1 Single - MIR-MRC (F-Net)	-	-	74.8	78.0
#2 Single - nlnet	-	-	74.2	77.1
Published				
unet (Ensemble)	-	-	71.4	74.9
SLQA+ (Single)	-		71.4	74.4
Ours				
$\text{BERT}_{\text{LARGE}}$ (Single)	78.7	81.9	80.0	83.1

Table 3: SQuAD 2.0 results. We exclude entries that use BERT as one of their components.

SWAG

- The Situations with Adversarial Generations (SWAG)

  On stage, a woman takes a seat at the piano.

  She …

  a) sits on a bench as her sister plays with the doll.
  b) smiles with someone as the music plays.
  c) is in the crowd, watching the dancers.
  d) nervously sets her fingers on the keys.

System	Dev	Test
ESIM+GloVe	51.9	52.7
ESIM+ELMo	59.1	59.2
OpenAI GPT	-	78.0
$\text{BERT}_{\text{BASE}}$	81.6	-
$\text{BERT}_{\text{LARGE}}$	**86.6**	**86.3**
Human (expert)†	-	85.0
Human (5 annotations)†	-	88.0

Table 4: SWAG Dev and Test accuracies. †Human performance is measured with 100 samples, as reported in the SWAG paper.

- The only task-specific parameters is a vector $V \in \boldsymbol{R}^H$
- The probability distribution is the **softmax** over the four choices

Conclusions

- Unsupervised pre-training (pre-training language model) is increasingly adopted in many NLP tasks.
  - Google Search is applying BERT models for search queries for over 70 languages.
- Major contribution of the paper is to propose a deep bidirectional architecture from Transformer.
  - Advance state-of-the-art for many important NLP tasks.

Links

- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova.
- BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding Devlin et al., 2018 (Google AI Language). Presenter: Phạm Quang Nhật Minh NLP Researcher Alt Vietnam
- The Illustrated BERT, ELMo, and co. (How NLP Cracked Transfer Learning). Jay Alammar. http://jalammar.github.io/illustrated-bert/
- https://cs.uwaterloo.ca/~mli/cs886-002-2020.html
- Other resources: http://primo.ai/index.php?title=Bidirectional_Encoder_Representations_from_Transformers_(BERT)
- Programming:
  - TensorFlow code and pre-trained models for BERT: https://github.com/google-research/bert
  - PyTorch Pretrained Bert: https://github.com/huggingface/pytorchpretrained-BERT
  - BERT-pytorch: https://github.com/codertimo/BERTpytorch
  - BERT-keras: https://github.com/Separius/BERTkeras

Question?

Slides available on:
https://personal-sites.deakin.edu.au/~mohamedb/teaching.html

* Bert is a Muppet who appears in numerous skits on the popular children's television show of the United States, **Sesame Street**.